

# Difference Between calloc and malloc

[www.differencebetween.com](http://www.differencebetween.com)

## Key Difference - calloc vs malloc

In programming, it is necessary to store data. Data are stored in the [memory](#). These memory locations are known as variables. Each variable has a specific type. They can be integers, floats, doubles, characters etc. There are also data structures that can store a fixed-size sequential collection of elements of the same type. It is an array. The programmer has to declare the array size. If the programmer declares an array of integers for five elements, it is not possible to assign a value to an index higher than the declared size. The memory allocation is fixed, and it cannot be changed at run time. Other memory allocation method is [dynamic memory allocation](#). Dynamic memory allocation helps to allocate more memory when required and release when necessary. The `<stdlib.h>` [header file](#) has four functions for dynamic memory allocation. `calloc` and `malloc` are two such functions. The **key difference** between `calloc` and `malloc` is that **`calloc` allocates the memory and also initialize the allocated memory blocks to zero whereas `malloc` allocates the memory but does not initialize that allocated memory to zero.** Accessing the content in `calloc` will give zero, but `malloc` will give a garbage value.

## What is calloc?

Memory allocation is the process of assigning memory for the executing programs. Sometimes it is necessary to change the size of memory. Therefore, dynamic memory allocation is used. It is done using [pointers](#). Pointers are reference variables that hold the address of another variable.



**Figure 01: calloc and malloc**

`calloc` stands for “contiguous allocation”. It allocates multiple blocks of memory with the same size. The syntax for `calloc` is as follows. It takes two arguments. They are the

number of blocks and the size of each block. The function `calloc` returns a void pointer, so a cast operator is used to returned pointer type according to the required data type.

```
void * calloc( size_t num, size_t size);
```

Refer the below simple C program.

```
#include< stdio.h>

#include <stdlib.h>

int main(){

int ptr* = (int *) calloc(20, sizeof(int));

if (ptr == NULL){

printf("Memory is not allocated");

}

else{

printf("Memory is allocated");

}

return 0;

}
```

According to the above program, a contiguous block of memory which can hold 20 elements is allocated. Each will have the size of an integer. The `sizeof(int)` is used because the integer type varies from compiler to compiler.

If the memory allocation is successful, it will return the base address of the memory block. It means that pointer `ptr` is now pointing to the base address of that memory block. All allocated regions are initialized to zeros. It will print the Memory Allocated message. If the memory allocation is unsuccessful, it will return the null pointer. Therefore, it will print Memory is not allocated message.

## What is malloc?

The `malloc` function is used to allocate the required amount of bytes in memory. The syntax for `malloc` is as follows. The size represents the required memory in bytes.

```
void *malloc(size_t_size);
```

The function malloc returns a void pointer, so a cast operator is used to returned pointer type according to the required data type.

Refer the below simple C program with malloc function.

```
#include< stdio.h>

#include <stdlib.h>

int main(){

int ptr* = (int *) malloc (10 * sizeof(int));

if (ptr == NULL){

printf("Memory is not allocated");

}

else{

printf("Memory is allocated");

}

return 0;

}
```

According to the above program, block of memory will be allocated. The pointer is pointing to the starting address of the allocated memory. The returned pointer is converted to an integer type. If memory is allocated it will print memory is allocated message. If the memory is not allocated, a null pointer will return. Therefore, memory is not allocated message will print.

## **What are the Similarities Between calloc and malloc?**

- Both these functions are declared in the header file <stdio.h>. It is the standard library header file.
- Both functions are used for dynamic memory allocation.
- The pointer returned by calloc and malloc should be casted into the specific type.
- In successful memory allocation, both functions will return a pointer with the base address of the memory block.

- If the memory allocation is unsuccessful, a null pointer will be returned.

## What is the Difference Between calloc and malloc?

calloc vs malloc	
calloc is a function for dynamic memory allocation in <a href="#">C language</a> stdlib.h header file that allocates a specific number of bytes and initializes them to zero.	malloc is a function for dynamic memory allocation in C language stdlib.h header file that allocates a specific number of bytes.
Meaning	
calloc stands for contiguous allocation.	malloc stands for memory allocation.
Syntax	
calloc follows a syntax similar to void *calloc(size_t_num, size_t_size);	malloc follows a syntax similar to void *malloc(size_t_size);.
Number of Arguments	
calloc takes two arguments. They are a number of blocks and size of each block.	malloc takes one argument. It is a number of bytes.
Speed	
calloc takes little longer than malloc. That is because of the extra step of initializing the allocated memory by zero.	malloc is faster than calloc.

## Summary - calloc vs malloc

In static memory allocation such as using arrays, the memory is fixed. If few elements are stored, then the rest of the memory is wasted. It might also cause errors when the allocated memory is small than the required memory. Therefore, dynamic memory allocation is used. In C language, calloc and malloc provide dynamic memory allocation. The difference between calloc and malloc is that calloc allocates memory and also initialize the allocated memory blocks to zero while malloc allocates the memory but does not initialize memory blocks to zero. Malloc takes two arguments while calloc takes two arguments.

### Reference:

- 1.Kumar, Krishan. "Difference Between Malloc and Calloc in C." Cs-Fundamentals.com, Cs-Fundamentals.com. [Available here](#)
- 2."C dynamic memory allocation." Wikipedia, Wikimedia Foundation, 13 Jan. 2018. [Available here](#)
- 3."Calloc() versus malloc()." GeeksforGeeks, 14 June 2017. [Available here](#)

## How to Cite this Article?

APA: Difference Between calloc and malloc.(2018 January 30). Retrieved (date), from <http://differencebetween.com/difference-between-calloc-and-vs-malloc/>

MLA: "Difference Between calloc and malloc" Difference Between.Com. 30 January 2018. Web.

Chicago: "Difference Between calloc and malloc." Difference Between.Com. <http://differencebetween.com/difference-between-calloc-and-vs-malloc/> accessed (accessed [date]).



Copyright © 2010-2017 Difference Between. All rights reserved